COMPARATIVE ANALYSIS FOR PERFORMANCE MEASUREMENTS OF SOFTWARE TESTING BETWEEN MOBILE APPLICATIONS AND WEB APPLICATIONS

ZAINAB HASSAN MUHAMAD

A dissertation submitted in
fulfilment of the requirements for the award of the
Degree of Master of Computer Science (Software Engineering)

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

JULY 2015

DEDICATION

This dissertation is dedicated with the deepest gratitude to my beloved parents,

Who has supported me and never gave up on me.

To my beloved husband,

To my dear brothers and sisters,

For all persons who has brighten my way to success.

May God bless us always.

ACKNOWLEDGEMENT

First and foremost, I thank God for the strength and courage that made this humble effort a reality. Then, I really was very grateful and would like to take this opportunity to say thank you very much to my parents for their love and support throughout my life. Thank you both for giving me strength to reach for the stars and chase my dreams. My special gratitude to my beloved husband, my dear son, and also to my brothers and sisters for their support.

I would like to extend my sincere thanks and appreciation to my supervisor Prof. Dr. Rosziati Ibrahim for his guidance, encouragement, and valuable comments throughout this study. Also, I would like to express my deepest gratitude to Universiti Tun Hussein Onn Malaysia (UTHM), and especially the Faculty of Computer Science and Information Technology / Department of Software Engineering for giving me the opportunity to further my Master's degree. To all my friends, thank you for your understanding and encouragement in many moments of crisis. Your friendship makes a wonderful experience. I cannot list all the names here, but you are always on my mind.

Last but not least, thanks to all persons and individuals who have been involved towards the completion of this dissertation, directly or indirectly.

ABSTRACT

Software testing has an important role in software engineering, and is fundamental to Software Quality Assurance (SQA). Besides the popularity of web applications, mobile applications have gained paralleled advancement despite increasing complexity. On one hand, this issue reflects the rising concerns for ensuring performance both of web and mobile applications. On the other hand, a comparative analysis of software testing issues between web and mobile applications has not been completed. Thus, this study aims to employ an effective testing approach that is able to adapt both of web and mobile application testing to detect possible failures. To achieve this, UML activity diagrams were developed from four case studies for web and mobile applications to describe the behaviour of those applications. Test cases were then generated by using the MBT technique from the developed UML activity diagrams. Performance measurements Hits per Second, Throughput and Memory Utilization for each case study were evaluated by execution of test cases that were generated by using HP LoadRunner 12.02 tool. Finally, the MSE of performance measurements was compared and analysed among the four case studies. The experimental results showed that the disparity between the mobile applications and web applications was obvious. Based on the comparison analysis for software testing of mobile applications versus web applications that was the web applications were lesser than mobile applications for software testing of four case studies in terms each of the Hits per Second, Throughput and Memory Utilization. Consequently, mobile applications need more attention in the testing process.

ABSTRAK

Ujian perisian mempunyai peranan penting dalam bidang kejuruteraan perisian, dan merupakan asas kepada Jaminan Kualiti Perisian (SQA). Selain aplikasi web yang semakin dikenali ramai, aplikasi mudah alih juga semakin digemari ramai walaupun menimbulkan sedikit kerumitan. Dari satu sudut, isu ini menimbulkan sedikit kebimbangan untuk memastikan kebolehan prestasi kedua-dua web dan aplikasi mudah alih ini. Walaubagaimanapun, analisis perbandingan isu untuk ujian perisian antara web dan aplikasi mudah alih telah dilakukan tetapi masih belum selesai. Oleh sebab itu, kajian ini bertujuan untuk membuat ujian keberkesanan yang mampu disesuaikan untuk kedua-dua aplikasi iaitu web dan aplikasi mudah alih untuk mengesan kegagalan mungkin. Untuk mencapai matlamat ini, gambar rajah aktiviti UML telah diuji ke atas empat kajian kes iaitu untuk web dan juga untuk aplikasi mudah alih yang mana menerangkan sifat-sifat ataupun cara kerja aplikasi tersebut. Kes-kes kajian kemudiannya telah dihasilkan dengan menggunakan teknik MBT daripada perkembangan aktiviti rajah UML. Ukuran prestasi *Hits per Second*, *Throughput* dan *Memory Utilization* untuk setiap kes kajian telah dinilai dengan menjalanakan ujian yang dihasilkan dengan menggunakan alat HP LoadRunner 12.02. Akhirnya, bacaan prestasi MSE yang telah dilakukan telah dibandingkan dan dianalisis di antara empat kes kajian yang lain. Keputusan eksperimen menunjukkan perbezaan di antara aplikasi mudah alih dan aplikasi web adalah sangat jelas. Berdasarkan analisis perbandingan untuk ujian perisian aplikasi mudah alih dan aplikasi web, keputusan menunjukkan aplikasi web adalah lebih rendah daripada aplikasi mudah alih untuk ujian perisian daripada empat kes kajian sama ada dari segi *Hits per Second*, *Throughput* dan *Memory Utilization*. Oleh itu, aplikasi mudah alih adalah lebih memerlukan perhatian di dalam proses pengujian.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

SQA	-	Software Quality Assurance
SUT	-	System Under Test
SDLC	-	Software Development Life Cycle
GUI	-	Graphical User Interface
MBT	-	Model Based Testing
FSM	-	Finite State Machine
UML	-	Unified Modelling Language
QA	-	Quality Assurance
OS	-	Operating System
iOS	-	iPhone OS
GPS	-	Global Positioning System
HTML	-	Hypertext Markup Language
SQL	-	Structured Query Language
EDS	-	Event-Driven Systems
API	-	Application Programming Interface
I/O	-	Input/Output
CSS	-	Cascading Style Sheets
PHP	-	Personal Home Page
ASP	-	Active Server Pages
JSP	-	Java Server Pages

WWW	-	World Wide Web
SBST	-	Search Based Software Test case generation
FBA	-	Flow Based Approach
SBT	-	Search Based Testing
UCTM	-	Use Case Transition Model
TPG	-	Test Paths Generating Tool
LTS	-	Labelled Transition Systems
MSC	-	Message Sequence Charts
OO	-	Object-Oriented
HTTP	-	Hypertext Transfer Protocol
HP	-	Hewlett Packard
MSE	-	Mean Square Error
TCBAD	-	Test Case Generation based on Activity Diagram
ADT	-	Activity Dependency Table
ADG	-	Activity Dependency Graph
UI	-	User Interface
CC	-	Cyclomatic Complexity
VuGen	-	Virtual user Generator

LIST OF APPENDIXES

CHAPTER 1

INTRODUCTION

1.1 Research Background

Software testing has an important role in software engineering, and it is fundamental to Software Quality Assurance (SQA). The objective of software testing is to show the differences between the expected and actual behaviours of the system under test (SUT). The goal of software testing is to detect whether the behaviour of the system implemented has visible differences from the expected behaviour stated in the specification (Sumit & Narendra, 2014).

A web application is a distributed interactive system, which provides a new view for users to deploy software applications. Web applications are based on multi architecture. With the popularity of web applications, the reliability and quality of web applications have become a very critical problem so they require testing in each and every phase but in an automated manner to get accurate results. Software testing is a primary way of improving software reliability and assuring software quality. Testing is the major phase while developing software because it makes the software much better by finding errors and making several improvements in the system. Testing web applications is a very important process as they are the fastest growing area in software engineering which provides several activities such as many business transactions, academic studies, etc. (Arora & Sinha, 2012; Song & Chen, 2012). Along with the fast growing market of smart mobile devices such as smartphones and tablet computers, the availability and popularity of smartphone applications have dramatically increased. Since more and more people are relying upon smartphone applications to manage their bills, schedules, emails, shopping, and so on, it is required that smartphone applications be user-friendly and reliable.

The development of the mobile application is based on Software Development Life Cycle (SDLC). Testing is essential for a software development lifecycle that impacts the popularity of any software and hardware products (Ang *et al*., 2014). Moreover, this trend has prompted an explosive growth in the number and variety of mobile applications being developed. Thus, developers are required to develop high quality applications in order to be competitive. On the other hand, mobile applications are usually developed in relatively small-scale projects which may not be able to support extensive and expensive manual testing. Thus, it is particularly important to develop automated testing tools for mobile applications (Yang *et al.,* 2013). The dynamics of web applications have motivated different testing approaches to be developed and introduced by researcher's numerous approaches to testing web applications. Naturally, the main aim of these test approaches is to discover failures in the required functionality, in order to verify the user behaviour with the Graphical User Interface (GUI) of the application (Suhaila & Wan, 2011). Similarly, under the increasing complexity and time-to-market pressures, performance validation is becoming a major issue for mobile applications. Due to the GUI intensive nature, the execution of mobile applications heavily relies on the user interactions with GUI, and few existing testing approaches are effective and automatically utilize GUI (Ang *et al*., 2014).

Testing is a very costly and time consuming process. So, in order to cut down on costs, save time, and increase reliability, Model Based Testing (MBT) approach were used in this study. MBT is a process of generating test cases and evaluating test results based on the design and analysis of models. Recently, MBT has gained attention with the popularization of modelling in software development. Several of software models are useful among others, such as Finite State Machine (FSM) and Unified Modelling Language (UML). FSM Models have a long history in design and testing, however the complex applications imply large state machines that are difficult to construct and maintain. The UML modelling based testing approach intends to solve this problem (Sandeep *et al.*, 2012; Ang *et al*., 2014). UML has now become the de facto standard for object oriented modelling and design. UML models are an important source of information for test case design, which, if satisfactorily exploited, can go a long way in reducing testing cost and effort and at the same time improving software quality. UML based automatic test generation is a practically important and theoretically challenging topic and is receiving increasing attention

from researchers (Santosh & Durga, 2010). MBT is one of the most of significant techniques that have been applied to generate test cases by using UML diagrams for web applications by (Zhang *et al.*, 2007; Ke *et al.*, 2010; Prachet & Abhishek, 2013). Moreover, MBT has been applied to mobile applications by (Chouhan *et al.*, 2012; Tobias & Volker, 2014; Ang *et al*., 2014).

Currently, UML activity diagrams support GUI modelling, automated test case generation and error diagnosis. This approach can reduce the overall test time, and can effectively detect fatal faults in mobile applications (Ang *et al.*, 2014). In addition, the UML activity diagram is one of the most important diagrams among the thirteen diagrams. It is characterized by the high level of abstraction compared to other diagrams like sequence diagrams, class diagrams, etc. Furthermore, it is able to represent loops and concurrent activities. UML activity diagrams capture the key system behaviour. Besides this, they are used for business modelling, control and object flow modelling, complex operation modelling, etc. The main advantage of this model is its simplicity and ease of understanding the flow of logic of the system as well. For all these reasons, activity diagrams are well suited for treating system level testing of web applications (Aye & Myat, 2014).

1.2 Problem Statement

Currently, mobile applications have parallel advancement with web applications. This issue reflects the rising concerns for ensuring performance both of web and mobile applications (Maryam & Rosziati, 2014). As the growth both of web and mobile applications is rapid, this issue was interesting to some researchers (Vikas & Rajesh, 2014; Prachet & Abhishek, 2013), and they have taken into consideration web application testing. On the other hand, other researchers (Tobias & Volker, 2014; Ang *et al.*, 2014) were focused on mobile application testing. But, the comparative analysis between web application testing and mobile application testing is an issue that has not yet been resolved. Thus, the motivation of this research is to employ an effective testing approach, which is able to adapt with both web and mobile application testing to discover failures in the required performance.

Therefore, the UML activity diagrams will be developed from four case studies for web and mobile applications to describe the behaviour of those applications. Test cases will then be generated using the MBT technique based on

Test Case Generation based on Activity Diagram (TCBAD) model from the developed UML activity diagrams. In addition, performance measurements Hits per Second, Throughput and Memory Utilization for each case study were evaluated by execution of test cases that were generated by using HP LoadRunner tool. Finally, the performance measurements Hits per Second, Throughput and Memory Utilization will be compared and analysed among the case studies.

1.3 Research Objectives

The main objectives of this study are to:

(i) Develop UML activity diagrams from four case studies for web and mobile applications in order to describe the behaviour of those applications.

(ii) Generate test cases by using the MBT technique from activity diagrams developed in (i) for each case study.

(iii) Evaluate performance measurements (Hits per Second, Throughput and Memory Utilization) for each case study by execution test cases generated in (ii) by using HP LoadRunner tool.

(iv) Compare and analyse the results obtained in (iii) based on performance measurements (Hits per Second, Throughput and Memory Utilization).

1.4 Research Scope

This study focuses on the MBT technique based on TCBAD model proposed by Chouhan *et al*. (2012) via modelling four case studies for web and mobile applications by using UML activity diagrams to describe the behaviour of those applications. The EdrawMax 7.9 tool (Edraw, 2014) is used to construct the activity diagram in the context of both web applications and mobile applications. This study concentrates on automation testing by using the HP LoadRunner 12.02 tool (HP LoadRunner, 2015) to evaluate the application performance in terms of Hits per Second, Throughput and Memory Utilization. For the intents of this study, performance evaluation regarding with web applications has been tested on Windows

8. On the other hand, performance evaluation of mobile applications has been tested on Android 4.2.2. The four selected case studies are the following:

(i) Inktera Books is an e-book store application as an Android application allows user to register, sign in, search for e-books by author name or e-book title, browse e-books by categories, and access to own library. It is available at Google play apps.

(ii) Kobo Books is an e-book store application as an Android application allows user to register, sign in, search for e-books by author name or e-book title, browse e-books by categories, and access to own library. It is available at Google play apps.

(iii) Inktera web-based app is an online e-book store allows users to register, sign in, search for e-books by author name or e-book title, browse e-books by categories, and access to own library. It is available at https://www.inktera.com.

(iv) Kobo web-based app is an online e-book store allows user to register, sign in, search for e-books by author name or e-book title, browse e-books by categories, and access to own library. It is available at http://www.kobobooks.com.

1.5 Dissertation Outline

This dissertation provides a description of and report on the effort that was carried out throughout the duration of the project in order to achieve the project scope and objectives. This dissertation is divided into five chapters that cover the whole research. Chapter 1 provides a brief introduction to software testing and its role in improving reliability and assuring the quality of mobile and web applications. Furthermore, this chapter explains the problem statement, objectives and scope of this project. Chapter 2 introduces the different views of the authors through an overview of the literature relating to testing for mobile applications and web applications. In this chapter, previous methods and techniques are discussed. In addition, related topics and works relevant to the study based on various journals and publications are reviewed and summarized. Chapter 3 contains a discussion of the research framework. The sections in this chapter cover the framework with attached

REFERENCES

Agarwal, B., Tayal, S. and Gupta, M. (2010). *Software Engineering and testing An Introduction.* Hingham, Toronto: Infinity Science Press, Jones and Bartlett's books. ISBN: 978-1-934015-55-1.

Amalfitano, D., Fasolino, A., Tramontana, P. and Amatucci, N. (2013). Considering Context Events in Event-based Testing of Mobile applications. *Proceedings of the 2013 IEEE Sixth International Conference on Software Testing. Verification and Validation workshops (ICSTW).* March 18-22, 2013. Luxembourg: IEEE.126-133.

Amalfitano, D., Fasolino, A., Tramontana, P., De Carmine, S. and Memon, A. (2012). Using GUI ripping for automated testing of Android applications. *Proceedings of the 27th ACM International Conference on Automated Software Engineering.* New York, NY, USA: ACM. 258– 261.

Ang, L., Zishan, Q., Mingsong, C. and Jing, L. (2014). ADAutomation: An Activity Diagram Based Automated GUI Testing Framework for Smartphone Applications. *Proceedings of the Eighth IEEE International Conference on Software Security and Reliability (SERE), June 30, 2014-July 2, 2014.* San Francisco, CA: IEEE. 68 – 77.

Anju, B. (2014). A Comparative Study of Software Testing Techniques. *International Journal of Computer Science and Mobile Computing,* 3(6), 579-584.

Arora, A. & Sinha, M. (2012). Web Application Testing. *International Journal of Scientific & Engineering Research,* 3(2), 1-6, ISSN 2229-5518.

Aye, K. & Myat, M. (2014). An Efficient Approach for Model Based Test Path Generation. *International Journal of Information and Education Technology,* 5(10), 763- 767. doi: 10.7763/IJIET.2015.V5.607.

Belli, F., Beyazit, M. and Memon, A. (2012). Testing is an Event-Centric Activity. *Proceedings of the IEEE Sixth International Conference on Software Security and Reliability Companion (SERE-C), 20-22 June 2012*. Gaithersburg, MD: IEEE. 198 - 206.

Ben (2014). *Characterizing and Measuring Hits per Second for Web Applications*. Retrieved on December 12, 2014, from http://blog.nexcess.net/2011/06/29/characterizing-and-measuring-Hits per Second-for-web-applications-2/.

Bhupendra, S. & Shashank, S. (2014). A Model For Performance Testing Of Ajax Based Web Applications. *International Journal of Research in Engineering and Technology (Ijret),* 3(4), 889- 893.

Bipin, P. & Rituraj, J. (2014). Importance of Unified Modelling Language for Test Case Generation in Software Testing. *International Journal of Computer Technology & Applications.*, 5 (2), 345-350. ISSN: 2229-6093.

Cartaxo, E., Neto, F. and Machado, P. (2007). Test Case Generation By Means Of UML Sequence Diagrams and Labeled Transition Systems. *Proceedings of the ISIC. IEEE International Conference on Systems, Man and Cybernetics, 7-10 Oct. 2007.* Montreal: IEEE. 1292 – 1297.

Chan, K., Chen, T. and Towey, D. (2002). Restricted random testing. *Proceedings of the 7th International Conference on 7th European Conference on Software Quality Helsinki, Finland, June 9–13, 2002,* London, UK: Springer. 321–330.

Charland, A. & Leroux, B. (2011). Mobile application development: Web vs. Native. *Magazine Communications of the ACM,* 54(5), 49-53. doi:10.1145/1941487.1941504.

Chen, M., Mishra, P. and Kalita, D. (2008). Coverage-driven automatic test generation for UML activity diagrams. *Proceedings of the ACM 18th Great Lakes symposium on VLSI (GLSVLSI).* New York, NY, USA: ACM. 139–142.

Chen, M., Qin, X. and Li, X. (2006). Automatic test case generation for UML activity diagrams. *Proceedings of the 2006 international workshop on Automation of software test (AST).* New York, NY, USA: ACM. 2–8.

Chen, M., Qin, X., Xu, W., Wang, L., Zhao, J. and Li, X. (2009). UML Activity Diagram-Based Automatic Test Case Generation for Java Programs. *The Computer Journal,* 52(5), 545–556, 2009. doi: 10.1093/comjnl/bxm057.

Chen, T., Kuo, F., Robert, G. and Tse, T. (2010). Adaptive Random Testing: The ART of test case diversity. *Journal of Systems and Software archive,* 83(1), 60-66. doi:10.1016/j.jss.2009.02.022.

Cheng, B., Xiaoyan, W., Xiaoxiao, H., Chen, J., (2011). Multimedia Conferencing Management Using Web Services Orchestration over Public Networks. *Journal of Convergence Information Technology (JCIT),* 6(6), 361 - 375.

Chouhan, C., Shrivastava, V. and Parminder, S. (2012). Test Case Generation based on Activity Diagram for Mobile Application. *International Journal of Computer Applications*, 57(23), 4-9. doi: 10.5120/9436-3563.

Chouhan, C., Shrivastava, V., Parminder, S. and Soni, P. (2013). Test Case Generation on the Origin of Activity Diagram for Navigational Mobiles. *International Journal of Advanced Computational Engineering and Networking,* 1(2), 32-36. ISSN (p): 2320-2106.

Cramer, H., Rost, M., Belloni, N., Bentley, F. and Chincholle, D. (2010). Research in the large. Using app stores, markets, and other wide distribution channels in Ubicomp research. *Proceedings of the ACM 12th International Conference Adjunct Papers On Ubiquitous Computing – Adjunct.* New York, NY, USA: ACM. 511-514.

Cuixiong, H. & Iulian, N. (2011). Automating GUI testing for Android applications. *In Proceedings of the ACM 6th International Workshop on Automation of Software Test (AST '11).* New York, NY, USA: ACM. 77-83. doi:10.1145/1982595.1982612.

Edraw (2014). *Edraw Max Pro.* Retrieved on October 09, 2014, from https://www.edrawsoft.com/EDrawMax.php.

Fan, X., Shu, J., Liu, L. and Liang, Q.J. (2009). Test Case Generation from UML Subactivity and Activity Diagram. *Proceedings of the IEEE 2nd International Symposium on Electronic Commerce and Security, 22-24 May 2009.* Nanchang: IEEE. 244–248.

Gahran, A. (2011). *What'S A Mobile App?*. Retrieved on October 02, 2014, from http://www.contentious.com/2011/03/02/whats-a-mobile-app/.

Gaurav, S. & Kestina, R. (2013). Software Testing Techniques for Test Cases Generation. *International Journal of Advanced Research in Computer Science and Software Engineering,* 3(9), 261-265.

Gouri, S. & Meetu, A. (2012). Software Performance Engineering. *International Journal of Research in IT & Management,* 2(2), 364-371. ISSN 2231-4334.

Guangzhu, J. & Shujuan, J. (2009). A Quick Testing Model of Web Performance Based on Testing Flow and its Application. *Proceedings of the IEEE 6th International Conference on Web Information Systems and Applications, 18-20 Sept. 2009.* Xuzhou, Jiangsu: IEEE. 57-61. doi:10.1109/WISA.2009.16.

HP LoadRunner (2015). *HP LoadRunner 12.02*. Retrieved on March 01, 2015, from http://www8.hp.com/us/en/software-solutions/LoadRunner-load-testing/index.html.

Huang, Y., Cao, J., Jin, B., Tao, X. and Lu, J. (2010). Cooperative cache consistency maintenance for pervasive internet access. *Wireless Communications and Mobile Computing,* 10(3), 436-450. doi:10.1002/wcm.819.

IBM (2012). *Native, web or hybrid mobile-app development.* IBM Software, Thought Leadership White Paper. Retrieved on September 10, 2014, from http://www.computerworld.com.au/whitepaper/371126/native-web-or-hybrid-mobile-app-development/download/.

IEEE (2004). *Guide to the Software Engineering Body of Knowledge*. United States of America: Angela Burgess. ISBN 0-7695-2330-7.

Imran, A. & Roopa, S. (2012). Quality Assurance And Integration Testing Aspects In Web Based Applications. *International Journal of Computer Science*

Engineering and Applications (IJCSEA), 2(3), 109-116. doi: 10.5121/ijcsea.2012.2310.

Imran, A., Gias, A.U. and Sakib, K. (2012). An Empirical Investigation of Cost-Resource Optimization for running Real-Life Applications in Open Source Cloud. *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS), 2-6 July 2012*. Madrid: IEEE. 718 - 723. doi:10.1109/HPCSim.2012.6267002.

Kalpan, B. & Ramakanth, K. (2012). A Survey on Performance Testing Approaches of Web Application and Importance of WAN Simulation in Performance Testing .*International Journal on Computer Science and Engineering (IJCSE),* 4(5), 883-885. ISSN: 0975-3397.

Kansomkeat, S., Offutt, J., Abdurazik, A. and Baldini, A. (2008). A Comparative Evaluation Of Tests Generated From Different UML Diagrams. *Proceedings of the IEEE 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 6-8 Aug. 2008.* Phuket: IEEE. 867–872. doi: 10.1109/SNPD.2008.48.

Kao, Y., Lin, C., Yang, K. and Yuan, S. (2011). A Cross-Platform Runtime Environment for Mobile Widget-Based Application *Proceedings of the IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 10-12 Oct. 2011.* Beijing: IEEE. 68 – 71. doi:10.1109/CyberC.2011.20.

Ke, H., Xiao-Hong, L. and Zhi-Yong, F. (2010). Approach to activity diagram model driven testing for Web applications. *Journal of Computer Applications,* 30(9), 2365-2369.

Khandai, M., Acharya, A. and Mohapatra, D.P. (2011). A Novel Approach of Test Case Generation for Concurrent Systems Using UML Sequence Diagram. *Proceedings of the IEEE 3rd International Conference on Electronics Computer Technology (ICECT), 8-10 April 2011*. Kanyakumari: IEEE.157–161. doi:10.1109/ICECTECH.2011.5941581.

Kim, H. (2013). Hybrid Model Based Testing for Mobile Applications. *International Journal of Software Engineering and Its Applications,* 7(3), 223-238.

Kulvinder, S. & Semmi (2013). Finite State Machine based Testing of Web Applications. *International Journal of Software and Web Sciences (IJSWS),* 4(1), 41- 46. ISSN: 2279-0063.

Li, L. Miao, H. and Qian, Z. (2008). A UML-Based Approach to Testing Web Applications. *Proceedings of the IEEE International Symposium on Computer Science and Computational Technology, 20-22 Dec. 2008.* Shanghai: IEEE. 397 – 401.

Liu, Z., Gao, X. and Long, X. (2010). Adaptive Random Testing of Mobile Application. *Proceedings of the IEEE 2nd International Conference on Computer Engineering and Technology, 16-18 April 2010.* Washington, DC, USA: IEEE. 297-301. doi: 10.1109/ICCET.2010.5485442.

Mario, K. & Eugene, O. (2014). *Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options.* Retrieved on September 20, 2014, from: https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options.

Maryam, A. & Rosziati, I. (2014). A Comparative Study of Web Application Testing and Mobile Application Testing. *Advanced Computer and Communication Engineering Technology, Lecture Notes in Electrical Engineering. Springer International Publishing,* 315, *491-500. doi: 10.1007/978-3-319-07674-4_48.*

Masi, E., Cantone, G., Mastrofini, M., Calavaro, G. and Subiaco, P. (2012). Mobile apps development: A framework for technology decision making. *Proceedings of International Conference on Mobile Computing, Applications, and Services.* Springer Berlin Heidelberg. 64–79. ISSN 1867-8211.

Mohd, E. & Farmeena, K. (2012). A Comparative Study of White Box, Black Box and Grey Box Testing Techniques. *International Journal of Advanced Computer Science and Applications,* 3(6), 12-15.

Mohd, E. (2010). Different Forms of Software Testing Techniques for Finding Errors. *IJCSI International Journal of Computer Science Issues*, 7(3), 11-16.

Mohd, E. (2011a). Different Approaches to Black Box Testing Technique for Finding Errors. *International Journal of Software Engineering & Application* (*IJSEA),* 2(4), 31-40.

Mohd, E. (2011b). Different Approaches to White Box testing Technique for Finding Errors. *International Journal of Software Engineering & Application (IJSEIA),* 5(3), *1-13.*

Myers (2004). *Art of Software Testing*. 3rd.Hoboken, NJ, USA: John Wiley & Sons, Incorporated.

Nadia, A. & Mark, H. (2011). Automated web application testing using search based software engineering. *Proceedings of the IEEE 26th International Conference on Automated Software Engineering (ASE), 6 - 10 November 2011*. Lawrence, Kansas, USA: IEEE. 3 – 12.

Nazish, R., Nadia, R., Saba, A. and Zainab, N. (2014). Model Based Testing in Web Applications. *International Journal of Scientific Engineering and Research (IJSER),* 2(1), 56-60.

Pakinam, N., Nagwa, L., Mohamed H. and Mohamed F. (2011a). A Proposed Test Case Generation Technique Based on Activity Diagrams. *International Journal of Engineering & Technology*, 11(3), 35-52.

Pakinam, N., Nagwa, L., Mohamed, H. and Mohamed, F. (2011b). Test Case Generation and Test Data Extraction Techniques. *International Journal of Electrical & Computer Sciences,* 11(3), 82-89.

Pallavi, R. & Satyaveer, T. (2014). Android Mobile Automation Framework. *International Journal of Engineering and Computer Science,* 3(10), 8555-8560.

Pavol, T., Ondrej, V., Lukas, S. (2012). The Usage of Performance Testing for Information Systems. *International Conference on Information and Computer Applications (ICICA),* 24 (12), 317- 321.

Pogue, D. (2009). *A Place to Put Your Apps.* Retrieved October 02, 2014, from: http://www.nytimes.com/2009/11/05/technology/personaltech/05pogue.html ?pagewanted=all.

Pourya, N., Suhaimi, B.,Mohammad, H. and Abolghasem, Z. (2013). A Comparative Evaluation of approaches for Web Application Testing. *The International Journal of Soft Computing and Software Engineering (JSCSE),* 3(3), 333-341. doi: 10.7321/jscse.v3.n3.50.

Prachet, B. & Abhishek, K. (2013). Model Based Regression Testing Approach of Service oriented Architecture (SOA) Based Application: A Case Study. *International Journal of Computer Science and Informatics,* 3(2), 11-16.

Prakash, V. & Gopala, S. (2012). Cloud Computing Solution- Benefits and Testing Challenges. *Journal of Theoretical and Applied Information Technology,* 39(2), 114-118. ISSN: 1992-8645.

Pratibha, F. & Manju, K. (2014). Research of Load Testing and Result Based on LoadRunner. *SSRG International Journal of Civil Engineering (SSRG-IJCE),* 1, 1-4. ISSN: 2348 – 8352.

Priya, S. & Sheba, P. (2013). Test Path Generation Using UML Sequence Diagram. *International Journal of Advanced Research in Computer Science and Software Engineering,* 3(4), 1069-1076.

Priyanka, J. & Mansi, G. (2014). Why Performance Testing. *International Journal of Advance Research in Computer Science and Management Studies,* 5(2), 45-49. ISSN 2222-1719.

Qin, X., Mishra, P. and Koo, H. (2012). *System-Level Validation: High-Level Modeling and Directed Test Generation Techniques*. Springer Science & Business Media, 2012. ISBN: 1461413583, 9781461413585.

Rahul, M. & Prince, J. (2013). Novel Testing Tools for a Cloud Computing Environment- A Review. *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA),* 1(3), 83-87. ISSN: 2321 – 2381.

Ranjita, K., Vikas, P. and Prafulla, K. (2013). Generation of Test Cases Using Activity Diagram. *International Journal of Computer Science and Informatics*, 3(2), 1-10. ISSN: 2231 –5292.

Ravi, R. (2013). Mobile Application Testing and Challenges. *International Journal of Science and Research (IJSR),* 2(7), 56-58.

Rosziati I. & Tan, S. (2009), Automatic Test Path Generation Based on UML Activity Diagram. *IEEE 5th International Conference on Computer, Telecommunication and System (ICTS 2009), Surabaya, Indonesia, August 2009.* 61-66. ISSN 2085-1944.

Rumbaugh, J., Jacoboson, I. and Booch, G. (2001). *The Unified Modeling Language User Guide*. Addison-Wesley.

Salkind, N. (2010). *Encyclopedia of Research Design.* SAGE. ISBN: 9781412961271.

Sandeep, K., Sangeeta, S. and Gupta, J. (2012). A Novel Approach for Deriving Test Scenarios and Test Cases from Events. *Journal of Information Processing Systems,* 8(2), 213-240. doi: 10.3745/JIPS.2012.8.2.213.

Santosh, K. & Durga, P. (2010). Test Case Generation from Behavioural UML Models. *International Journal of Computer Applications,* 6(8), 5-11.

Selvam, R. & Kartikeyani,V. (2011). Mobile Software Testing-Automated Test Case Design Strategies. *International Journal on computer Science and engineering (IJCSE).* 3(4), 1450-1461. ISSN: 0975-3397.

Shanthi, A. & Mohankumar, G. (2012). A Novel Approach for Automated Test Path Generation Using TABU Search Algorithm. *International Journal of Computer Applications, 48(13),* 222-224. ISSN 2250-0987.

Sheetal, S. & Joshi, S. (2012). Identification of Performance Improving Factors for Web Application by Performance Testing. *International Journal of Emerging Technology and Advanced Engineering,* 2(8), 433-436. ISSN 2250-2459.

Shilpa, S. & Meenakshi, S. (2014). Enhanced Grinder Framework with Scheduling and Improved Agents. *International Journal of Computer Science and Information Technologies,* 5(1), 55-59.

Shirole, M. & Kumar, R. (2013). UML Behavioural Model Based Test Case Generation: A Survey. *ACM SIGSOFT Software Engineering Notes,* 38(4), 1–13.

Shivani, A. & Vidhi, P. (2013). Bridge between Black Box and White Box – Gray Box Testing Technique. *International Journal of Electronics and Computer Science Engineering,* 2(1), 175-185. ISSN- 2277-1956.

Siy, M., Zhijie, Q. and Lei, L. (2010). Research on Mobile Web Applications End to End Technology. *Proceedings of the IEEE 10th International Conference on Computer and Information Technology (CIT), June 29 2010-July 1 2010.* Bradford: IEEE. 2061 – 2065. doi:10.1109/CIT.2010.350.

Song, B. & Chen, S. (2012). Coverage Criteria Guided Web Application Interactions Testing. *Proceedings of the IEEE 3rd World Congress on Software Engineering (WCSE), 6-8 Nov. 2012.* Wuhan: IEEE. 46 – 50. doi:10.1109/WCSE.2012.17.

Song, B., Gong, S. and Chen, S. (2011). Model Composition and Generating Tests for Web Applications. *In Seventh IEEE International Conference on Computational Intelligence and Security, 2011.*

Suhaila, M. & Wan, M. N. (2011). An Outlook of State-of-the-Art Approaches in Functional Testing of Web Application. *International MultiConference of Engineers and Computer Scientists,* 744 -749. ISBN 978-988182103-4.

Sumit, M. & Narendra, K. (2014). Model Based Testing Of Website. *International Journal on Computational Sciences & Applications (IJCSA),* 4(1), 143-152. doi:10.5121/ijcsa.2014.4114.

Swain, S.K., Pani, S.K. and Mohapatra, D.P. (2010). Model Based Object-Oriented Software Testing. *Journal of Theoretical & Applied Information Technology,* 14(1/2), 30-36.

Takala, T., Katara, M. and Harty, J. (2011). Experiences of system-level model-based GUI Testing of an Android application. *Proceedings of the IEEE 4th International Conference on Software Testing, Verification, and Validation (ICST 2011).* Los Alamitos, CA, USA: IEEE. Mar. 2011. 377–386.

Tarkoma, S. (2009). *Mobile Middleware-Architectures, Patterns, and Practice. UK: John Wiley & Sons.*

Thirumalai, R. & Balasubramanian, N. (2013). Performance Measurement of Web Applications Using Automated Tools. *Proceedings of the International MultiConference of Engineers and Computer Scientists, 13 – 15 March, 2013.* Hong Kong: IMECS. ISBN: 978-988-19251-8-3.

Tobias, G. & Volker, G. (2014). A Model-Based Approach to Test Automation for Context-Aware Mobile Applications. *Proceedings of the ACM 29th Annual Symposium on Applied Computing.* New York, NY, USA: ACM. 420-427. doi:10.1145/2554850.2554942.

Vikas, S. & Rajesh, B. (2014). Model based Test Cases Generation for Web Applications. *International Journal of Computer Applications (0975 – 8887),* 92(3), 23-31.

Vipin, S. & Ajay, P. (2012). Representation of Object-Oriented Database for the Development of Web Based Application Using Db4o. *Journal of Software Engineering and Applications,* 5(9), 687-694. doi: 10.4236/jsea.2012.59082.

Wang, L., Yuan, J., Yu, X., Hu, J., Li, X. and Zheng, G. (2004). Generating test cases from UML activity diagram based on gray-box method. *Proceedings of the IEEE 11th Asia-Pacific Software Engineering Conference (APSEC), 30 Nov.-3 Dec. 2004.* IEEE. 284–291. doi:10.1109/APSEC.2004.55.

Wasserman, A. (2010). Software Engineering Issues for Mobile Application Development. *Proceedings of the ACM FSE/SDP workshop on Future of software engineering research.* New York, NY, USA: ACM. 397–400. doi:10.1145/1882362.1882443.

Weibiao, Z., Fang, H., Shuanshuan, B., Lei, D., Jie, F., Ling, J. (2014). Performance Evaluation of Concurrent-access-intensive WebGIS Applications Based on

CloudStack. *International Journal of Advancements in Computing Technology(IJACT)*, 6(5), 41-52.

Yang, W., Prasad, M. and Xie, T. (2013). A Grey-Box Approach for Automated GUI-Model Generation of Mobile Applications. *Proceedings of the 16th International Conference, FASE 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS, March 16-24, 2013.* Rome, Italy: Springer. 250–265. doi:10.1007/978-3-642-37057-1_19.

Yasir, S. & Olvisa, D. (2011). Web application performance modeling using layered queuing networks. *Electronic notes in theoretical Computer science,* 275,123-142.

Zeng, F., Chen, Z., Cao, Q. and Mao, L. (2009). Research On Method Of Object-oriented Test Cases Generation Based on UML and LTS. *Proceedings of the IEEE 1st International Conference on Information Science and Engineering (ICISE), 26-28 Dec. 2009.* Nanjing: IEEE. 5055 – 5058. doi:10.1109/ICISE.2009.965.

Zhang, G., Rong, M. and Zhang, J. (2007). A Business Process of Web Services Testing Method Based on UML2.0 Activity Diagram. *IEEE Workshop on Intelligent Information Technology Application.* Zhang Jiajie: IEEE. 59- 65. doi:10.1109/IITA.2007.83.